Accordion Component Documentation

Overview

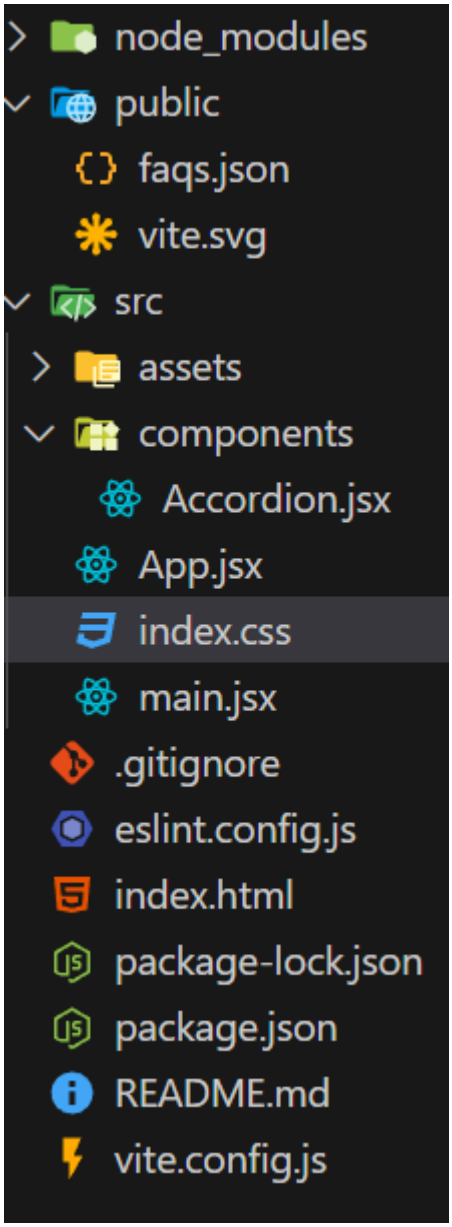
The **Accordion Component** is a React-based UI component that displays a list of Frequently Asked Questions (FAQs). Users can interact with the component by clicking on a question to expand its corresponding answer. Only one question can be expanded at a time, and clicking on another question will collapse the previously opened one. The component fetches FAQ data from a JSON file (faqs.json) and dynamically renders the questions and answers.

Features

1. **Dynamic Data Fetching**: The component fetches FAQ data from a JSON file (faqs.json) using the fetch API.
2. **Single Open Item**: Only one FAQ item can be expanded at a time. Clicking on another question will collapse the previously opened one.
3. **Toggle Functionality**: Clicking on an already expanded question will collapse it.
4. **Responsive Design**: The component is styled with CSS and is responsive across different screen sizes.
5. **Smooth Transitions**: The component includes smooth transitions for expanding and collapsing FAQ items.

File Structure

The project structure is as follows:



Key Files:

- **Accordion.jsx**: Contains the React component for the Accordion.
- **faqs.json**: Contains the FAQ data in JSON format.
- **index.css**: Contains the CSS styles for the Accordion component.

- **App.jsx**: The main application file where the Accordion component is rendered.

Component Breakdown

1. State Management

The component uses React's useState and useEffect hooks for state management and side effects.

- **faqs**: This state holds the list of FAQs fetched from the JSON file.
- **openId**: This state tracks the ID of the currently open FAQ item. If no item is open, it is set to null.

```
const [faqs, setFaqs] = useState([]);

const [openId, setOpenId] = useState(null);
```

2. Data Fetching

The useEffect hook is used to fetch the FAQ data from the faqs.json file when the component mounts.

```
useEffect(() => {
  fetch('/faqs.json')
    .then((response) => response.json())
    .then((data) => setFaqs(data.faqs))
    .catch((error) => console.error('Error fetching FAQs:', error));
}, []);
```

3. Event Handling

The handleClick function is responsible for toggling the open state of the FAQ items. If the clicked item is already open, it will be closed. Otherwise, the clicked item will be opened, and any previously open item will be closed.

```
const handleClick = (id) => {
  setOpenId((prevId) => (prevId === id ? null : id));
```

```
};
```

4. Rendering

The component renders the list of FAQs using the map function. Each FAQ item is conditionally styled based on whether it is open or closed.

```
return (
  <div className="accordion">
    {faqs.map((faq) => (
      <div
        key={faq.id}
        className={`accordion-item ${openId === faq.id ? 'active' : ''}`}
      >
        <div
          className="accordion-question"
          onClick={() => handleClick(faq.id)}
        >
          {faq.question}
        </div>
        {openId === faq.id && (
          <div className="accordion-answer">{faq.answer}</div>
        )}
      </div>
    ))}
  </div>
);
```

5. Styling

The component is styled using CSS. The styles include:

- **Basic Layout**: The accordion is centered on the page with a maximum width of 800px.
- **Hover Effects**: The FAQ items have hover effects to improve user interaction.
- **Transitions**: Smooth transitions are applied to the expand/collapse animations.
- **Responsive Design**: The component is responsive and adjusts its layout for smaller screens.

```
.accordion {
  max-width: 800px;
  margin: 0 auto;
  padding: 20px;
  font-family: 'Arial', sans-serif;
}

.accordion-item {
  border: 1px solid #e0e0e0;
  margin-bottom: 10px;
  border-radius: 8px;
  overflow: hidden;
  background-color: #ffffff;
  box-shadow: 0 2px 8px rgba(0, 0, 0, 0.1);
  transition: box-shadow 0.3s ease;
}

.accordion-question {
  padding: 16px 20px;
```

```
  background-color: #f8f9fa;
  cursor: pointer;
  font-weight: 600;
  font-size: 18px;
  color: #333;
  display: flex;
  justify-content: space-between;
  align-items: center;
  transition: background-color 0.3s ease;
}

.accordion-answer {
  padding: 16px 20px;
  background-color: #ffffff;
  font-size: 16px;
  color: #555;
  line-height: 1.6;
  border-top: 1px solid #e0e0e0;
  display: none;
}

.accordion-item.active .accordion-answer {
  display: block;
}
```

6. Responsive Design

The component includes media queries to ensure it looks good on smaller screens.

```
@media (max-width: 768px) {
  .accordion {
    padding: 10px;
  }

  .accordion-question {
    font-size: 16px;
    padding: 14px 16px;
  }

  .accordion-answer {
    font-size: 14px;
    padding: 14px 16px;
  }
}

@media (max-width: 480px) {
  .accordion-question {
    font-size: 14px;
    padding: 12px 14px;
  }
  .accordion-answer {
    font-size: 13px;
    padding: 12px 14px;
  }
}
```

Usage

To use the Accordion component in your React application:

1. **Install Dependencies**: Ensure you have React and any necessary dependencies installed.
2. **Add faqs.json**: Place the faqs.json file in the public directory with the required FAQ data.
3. **Import Component**: Import the Accordion component into your App.jsx or any other parent component.
4. **Render Component**: Render the Accordion component in your application.

```
import React from 'react';

import Accordion from './components/Accordion';


function App() {
  return (
    <div className="App">
      <h1>FAQ Accordion</h1>
      <Accordion />
    </div>
  );
}


export default App;
```

Conclusion

The Accordion Component is a simple yet effective way to display a list of FAQs in a React application. It leverages React's state management and conditional rendering to provide a smooth user experience. The component is also styled with CSS to ensure it looks good across different devices and screen sizes.